

# The ProtégéVOWL Plugin: Ontology Visualization for Everyone

Steffen Lohmann<sup>1</sup>, Stefan Negru<sup>2</sup>, and David Bold<sup>1</sup>

Institute for Visualization and Interactive Systems (VIS),  
University of Stuttgart, Universitätsstr. 38, 70569 Stuttgart, Germany

`steffen.lohmann@vis.uni-stuttgart.de`

Faculty of Computer Science, Alexandru Ioan Cuza University  
Strada General Henri Mathias Berthelot 16, Iasi, Romania

`stefan.negru@info.uaic.ro`

**Abstract.** The Visual Notation for OWL Ontologies (VOWL) provides a visual language for the representation of ontologies. In contrast to related work, VOWL aims for an intuitive and interactive visualization that is also understandable to users less familiar with ontologies. This paper presents ProtégéVOWL, a first implementation of VOWL realized as a plugin for the ontology editor Protégé. It accesses the internal ontology representation provided by the OWL API and defines graphical mappings according to the VOWL specification. The information visualization toolkit Prefuse is used to render the visual elements and to combine them to a force-directed graph layout. Results from a preliminary user study indicate that ProtégéVOWL does indeed provide a comparatively intuitive and usable ontology visualization.

## 1 Introduction

Ontologies are not only the backbone of the Semantic Web, but they are increasingly used in many different contexts. More and more people in modern knowledge societies get in contact with ontologies as a means to structure and organize information. This requires a new generation of tools that enable not only experts but also non-experts to work with ontologies. Visualizations play a key role in this context, as they can assist users in reading and understanding ontologies.

While several visualizations for ontologies have been developed in the last couple of years, they either focus on specific ontology aspects or are hard to read for non-expert users. The silver bullet would be an ontology visualization that is equally comprehensive and comprehensible. It must be printable but also provide intuitive ways to interactively explore ontologies.

Having these goals in mind, we developed the Visual Notation for OWL Ontologies (VOWL) that provides a well-specified visual language for the user-oriented representation of ontologies. It defines graphical depictions for many elements of the Web Ontology Language (OWL) that are combined to a graph visualization representing the ontology. This paper presents ProtégéVOWL, a first implementation of VOWL realized as a plugin for the ontology editor Protégé.

## 2 Related Work

Many ontology visualizations focus on the representation of certain aspects of ontologies. Examples are OWLViz<sup>1</sup>, OntoGraf<sup>2</sup>, KC-Viz [13], OWLPropViz<sup>3</sup>, and the RelFinder [9] that visualize either the class hierarchy of ontologies or selected property relations. This is similar in approaches that use special types of diagrams to depict ontologies, such as treemaps [7,15], nested graphs [15], or cluster maps [6]. While these visualizations are partly very usable, they are limited to ontology aspects that can be expressed with the respective diagram types.

A very powerful type of diagram related to OWL and often reused to visualize ontologies is the class diagram of the Unified Modeling Language (UML). Examples of OWL tools that make use of UML class diagrams are the Visual Ontology Modeler (VOM)<sup>4</sup> or OWLGrEd [2]. Mappings between OWL elements and UML class diagrams are precisely defined in the Ontology Definition Metamodel (ODM) published by OMG<sup>5</sup>. However, while class diagrams are well known to many people working in IT, they are less familiar to people from other domains who usually find it difficult to interpret these diagrams [14].

Most closely related to VOWL are approaches that use graph visualizations to represent ontologies. Examples are TGViz [1], NavigOWL [10], GrOWL [11], and SOVA<sup>6</sup>. All these tools provide force-directed graph layouts resulting in appealing visualizations. TGViz, OntoGraf, and NavigOWL use simple notations where the visual elements look very similar, making the different OWL elements hardly distinguishable. In contrast, GrOWL and SOVA define more elaborated notations with different symbols, colors, and node shapes. However, they are designed for users with expertise in description logic and related symbols.

## 3 Visual Notation for OWL Ontologies (VOWL)

The main goal of VOWL is to provide a visual language that can also be understood by users less familiar with ontologies. We therefore avoided to use formal symbols to represent OWL constructs, like in GrOWL or SOVA. Instead, we designed visual elements that are more intuitive and easy to use. These visual elements are based on only a handful graphical primitives forming the alphabet of the visual language: Classes are depicted as circles that are connected by lines and arrows representing the property relations, while property labels and datatypes are shown in rectangles. The visual elements are combined to a graph visualization representing the ontology and being arranged in a force-directed layout.

<sup>1</sup> <http://protegewiki.stanford.edu/wiki/OWLViz>

<sup>2</sup> <http://protegewiki.stanford.edu/wiki/OntoGraf>

<sup>3</sup> <http://protegewiki.stanford.edu/wiki/OWLPropViz>

<sup>4</sup> <http://thematix.com/tools/vom/>

<sup>5</sup> <http://www.omg.org/spec/ODM/>

<sup>6</sup> <http://protegewiki.stanford.edu/wiki/SOVA>

Some OWL elements are treated in a special way to increase the readability of the visualization. For instance, the predefined classes `owl:Thing` and `rdfs:Resource` usually do not carry domain information. They are multiplied and depicted in smaller size in order to give them less prominence in the visualization. Similarly, `rdfs:datatype` and `rdfs:literal` are shown multiple times so that datatype properties are arranged radially around the classes they are connected with.

In addition, VOWL defines a color scheme for a better distinction of the different elements. The colors are defined in an abstract way leaving room for customization, but concrete colors and color codes are recommended by VOWL. However, colors are not required to use the visualization but it is also understandable when printed in black and white or read by color-blind people. Details that rely on color, such as the subtle distinction of `owl:Class` and `rdfs:Class` may be added as text information in these cases. The complete VOWL specification is publicly available at the persistent URL <http://purl.org/vowl/spec>.

## 4 ProtégéVOWL

ProtégéVOWL implements version 2.0 of the VOWL specification to a large extent. Like the specification, it focuses on the visualization of the ontology schema (i.e. the classes, properties and datatypes, sometimes called TBox), while it does not consider individuals and data values (the ABox) for the time being. ProtégéVOWL is based on Java and deployed as a JAR file that must be copied to the *plugins* folder of the Protégé installation. It is released under the MIT license and available for download at <http://vowl.visualdataweb.org>.

Figure 1 shows a screenshot of ProtégéVOWL depicting revision 1.35 of the SIOC Core Ontology [3]. The user interface consists of three parts: The *VOWL Viewer* displaying the ontology visualization, the *VOWL Sidebar* listing details about the selected element, and the *VOWL Controls* allowing to adapt the force-directed graph layout.

**VOWL Viewer:** ProtégéVOWL uses the visualization toolkit Prefuse [8] to render the visual elements and to arrange them in a force-directed graph layout. It accesses the internal ontology representation provided by the OWL API of Protégé and transforms it into the data model required by Prefuse. The OWL elements are mapped to the graphical representations as specified by VOWL and combined to a graph. Prefuse uses a physics simulation to generate the force-directed graph layout, consisting of three different forces: edges act as springs, while nodes repel each other and drag forces ensure that nodes settle [8]. The forces are iteratively applied resulting in an animation that dynamically positions the nodes. The user can smoothly zoom in to analyze certain ontology parts in detail or zoom out to explore the global structure of the ontology. They can pan the background and move elements around, which results in a repositioning of the nodes by an animated adaptation of the force-directed layout.

**VOWL Sidebar:** If an element is selected in the visualization, details about it are shown in the sidebar, such as for the selected class `sIOC:UserAccount` in

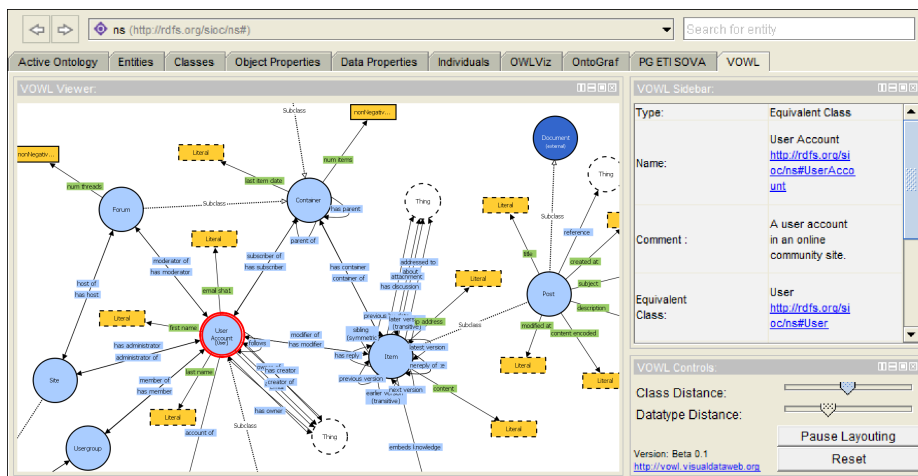


Fig. 1: ProtégéVOWL consists of the ontology visualization, a sidebar and control panel.

Figure 1. The details are provided by the OWL API and include its URI and comments added to the element, among others. URIs are displayed as hyperlinks that can be opened with a web browser or other tools for further exploration.

**VOWL Controls:** The repelling forces of classes and datatypes are adaptable with the two sliders in the VOWL Controls. This allows to fine-tune the force-directed layout in accordance with the size and structure of the visualized ontology. Since datatypes have their own repelling force, they can be positioned in close proximity to the classes they are connected with—to emphasize their radial arrangement and further increase readability. In addition, the layout animation can be paused via the VOWL Controls. This does not only reduce processor load but can also be useful to manually adapt, copy and print the visualization.

## 5 Evaluation and Future Work

We conducted a small user study in which we compared ProtégéVOWL with the SOVA plugin for Protégé, which is conceptually and technically most closely related to ProtégéVOWL (see Sec. 2). We recruited six participants that were familiar with the idea of ontologies but would not consider themselves ontology experts. We presented them both ProtégéVOWL and SOVA in alternating order. They had to solve 16 different tasks with the tools (eight with each) using two ontologies of different size (MUTO [12] and FOAF [5]). The tasks considered different aspects of the ontologies, such as determining the approximate number of classes or whether any class is equivalent to a given one. Finally, the participants had to rate both tools with regard to the perceived intuitiveness and ease-of-use.

All six participants preferred ProtégéVOWL over SOVA. They considered the VOWL notation to be more clear and liked the clean layout of the visualization. They also found it easier to distinguish different OWL elements and understand

how they are related. Overall, the results suggest that ProtégéVOWL does indeed provide a comparatively intuitive and usable ontology visualization that can also be understood by users without much expertise in ontologies. However, this needs to be further validated in a more comprehensive user study in the future.

Apart from that, we plan to integrate more features into ProtégéVOWL. In particular, we would like to address the representation of individuals and find more compact visualizations for very large ontologies. Furthermore, we are currently working on a web implementation of VOWL using JavaScript and SVG along with the visualization library D3 [4]. A first prototype is available together with ProtégéVOWL at <http://vowl.visualdataweb.org>.

## References

1. H. Alani. TGVizTab: An ontology visualisation extension for Protégé. In *Proc. VIKE '04*, 2003.
2. J. Bārzdīņš, G. Bārzdīņš, K. Čerāns, R. Liepiņš, and A. Sproģis. OWLGrEd: a uml style graphical notation and editor for OWL 2. In *Proc. OWLED '10*. CEUR, vol. 614, 2010.
3. U. Bojārs and J. Breslin. SIOC Core Ontology Specification. <http://rdfs.org/sioc/spec/>, 2010.
4. M. Bostock, V. Ogievetsky, and J. Heer. D3 data-driven documents. *IEEE Trans. Vis. Comput. Graph.*, 17(12):2301–2309, 2011.
5. D. Brickley and L. Miller. FOAF Vocabulary Specification 0.99. <http://xmlns.com/foaf/spec/>, 2014.
6. C. Fluit, M. Sabou, and F. van Harmelen. Ontology-based information visualization. In *Visualizing the Semantic Web*, pages 36–48. 2002.
7. F. J. García-Peñalvo, R. Colomo-Palacios, J. García, and R. Therón. Towards an ontology modeling tool. A validation in software engineering scenarios. *Expert Syst. Appl.*, 39(13), 2012.
8. J. Heer, S. Card, and J. Landay. Prefuse: A toolkit for interactive information visualization. In *Proc. CHI '05*, pages 421–430. ACM, 2005.
9. P. Heim, S. Lohmann, and T. Stegemann. Interactive relationship discovery via the semantic web. In *Proc. ESWC '10*, pages 303–317. Springer, 2010.
10. A. Hussain, K. Latif, A. Rextin, A. Hayat, and M. Alam. Scalable visualization of semantic nets using power-law graphs. *Appl. Math. Inf. Sci.*, 8(1):355–367, 2014.
11. S. Krivov, F. Villa, R. Williams, and X. Wu. On visualization of OWL ontologies. In *Semantic Web*, pages 205–221. Springer, 2007.
12. S. Lohmann, P. Díaz, and I. Aedo. MUTO: the modular unified tagging ontology. In *Proc. I-SEMANTICS '11*, pages 95–104. ACM, 2011.
13. E. Motta, P. Mulholland, S. Peroni, M. d’Aquin, J. M. Gomez-Perez, V. Mendez, and F. Zablith. A novel approach to visualizing and navigating ontologies. In *Proc. of ISWC '11*, pages 470–486. Springer, 2011.
14. S. Negru, F. Haag, and S. Lohmann. Towards a unified visual notation for OWL ontologies: Insights from a comparative user study. In *Proc. I-SEMANTICS '13*, pages 73–80. ACM, 2013.
15. M.-A. Storey, N. F. Noy, M. Musen, C. Best, R. Ferguson, and N. Ernst. Jambalaya: An interactive environment for exploring ontologies. In *Proc. IUI '02*, pages 239–239. ACM, 2002.